IN THE CLAIMS

Please amend the claims as follows:

Claim 1 (Currently Amended): A method of operating a computer having a pipelined processor having a Branch Target Buffer (BTB) comprising:

creating a recent entry queue ~~, said recent entry queue~~ comprising a small subset of ~~Branch Target Buffer (BTB)~~ entries in said BTB logically positioned in parallel with [[the]] said BTB; ~~branch target buffer (BTB), and~~

organizing ~~the~~ said recent entry queue as a First In First Out (FIFO) queue wherein when a new entry is placed into said recent entry queue, an oldest entry therein is moved out to make room for said new entry; [[,]]

organizing said BTB with a plurality of multi-associative classes with ~~branch target buffer (BTB) and~~ said recent entry queue being associative; [[and]]

defining said recent entry queue ~~being~~ logically ~~defined~~ as a subset of [[the]] said BTB ~~branch target buffer (BTB)~~ and coupled to track [[the]] a last number of branches entered into said BTB; ~~and also the~~

comparing each new entry to most recent entries into said recent entry queue ~~thereby allowing a comparison of recent entries of said recent entry queue to said BTB~~; and

[[for]] blocking duplicate entries from being installed into [[the]] said BTB and said recent entry queue by examining [[the]] contents of [[the]] said recent entry queue for such duplicate entries prior to a write into said BTB and into said recent entry queue; and in addition for allowing [[a]] decoding [[e]] to be delayed ~~stalled~~ by a defined ~~amount~~ number of cycles such that a branch of interest can be delayed from decoding in order to allow a given entry in [[the]] said BTB to be detected in time for future decoding [[es]] of said branch of interest.

Claims 2-7 (Canceled)

Claim 8. (Currently Amended) The method of claim 1 comprising searching [the]] said BTB for a next predicted branch and evaluating the recent entry queue while [the]] said BTB is being indexed.

Claim 9. (Currently Amended) The method of claim 8 wherein [the]]

said recent entry queue maintains a depth up to the associativity of [the]] said BTB;

whereby while the BTB is indexed, the recent entry queue positions are input to comparison logic.

Claim 10. (Currently Amended) The method of claim 8 comprising searching [the]] said recent entry queue for a matching branch in parallel to searching BTB output.

Claim 11. (Currently Amended) The method of claim 10 comprising creating hit detect logic to support the associativity of [the]] said BTB.

Claim 12. (Currently Amended): The method of claim 8 comprising using a subset of the recent entry queue as a subset of [[the]] said BTB.

Claim 13. (Originally Presented) The method of claim 12 comprising fast indexing recently encountered branches.

Claim 14. (Currently Amended): The method of claim 12 comprising:

providing a complete recent entry queue; and

searching [the]] said complete recent entry queue to block duplicate BTB writes.

Claims 15 -20 (Canceled)

21. (Originally Presented) The method of claim 1 comprising staging writes to the BTB in the recent entry queue.

22. (Previously presented) The method of claim 21 comprising delaying a write and placing the write in the recent event queue.

23. (Previously presented) The method of claim 22 comprising detecting a predicted branch while its BTB write is temporarily staged in the recent entry queue.

24. (Currently Amended) A computer having a pipelined processor comprising:

a comparator for comparing a Branch Target Buffer (BTB) with a recent entry queue; [[,]]

said recent entry queue comprising a set of branch target buffer (BTB) entries logically positioned in parallel with [[the]] said BTB; ~~branch target buffer (BTB),~~

said computer organizing [[the]] said recent entry queue as a FIFO queue wherein when a new entry is placed into said recent entry queue, an oldest entry therein is moved out to make room for said new entry;

said BTB being organized into multi-associative classes ~~branch target buffer (BTB)~~ and said recent entry queue being [[set]] associative; [[and]]

said recent entry queue being logically defined as a subset of [[the]] entries in said BTB~~-Branch Target Buffer (BTB)~~ and coupled to track [[the]] a last number of branches entered into said BTB; ~~and also [the~~

comparing each new entry to most recent entries into said recent entry queue; ~~thereby allowing a comparison of recent entries of said recent entry queue to said BTB~~; and [[for]]

said recent entry queue blocking duplicate entries from being installed into[[the]] said BTB and into said recent entry queue by examining- the contents of [[the]] said recent entry queue for such duplicate entries prior to a write into said BTB and said recent entry queue and in addition for allowing a decode to be delayed ~~stalled~~ by a defined ~~amount~~ number of cycles such that a branch of interest can be delayed from decoding- in order to allow a given entry in the BTB to be detected in time for future decodes of said branch of interest.

Claims 25- 26 (Canceled)

27. (Currently Amended) The computer of claim 24 [[26]] wherein the recent entry queue is fully associative for reading.

28. (Currently Amended) A program product comprising:

a computer readable medium having computer readable code thereon for controlling and configuring a computer having a pipelined processor and a Branch Target Buffer (BTB) to create a recent entry queue; [[,]]

said recent entry queue comprising a set of BTB ~~branch target buffer (BTB~~) entries logically positioned in parallel with said BTB; ~~the branch target buffer (BTB))~~,

organizing the recent entry queue as a FIFO queue wherein when a new entry is placed into said recent entry queue, an oldest entry therein is moved out to make room for said new entry; [[,]]

organizing said BTB into a plurality of associative classes ~~branch target buffer (BTB~~) and said recent entry queue being [[set]] associative; ~~and~~

definiing said recent entry queue being logically ~~defined~~ as a subset of [[the]] said BTB ~~branch target buffer (BTB~~) and coupled to track the last number of branches entered into said BTB; ~~and also the~~

comparing each new entry to most recent entries into said recent entry queue ~~thereby allowing a comparison of recent entries of said recent entry queue to said BTB and for~~

blocking duplicate entries from being installed into [[the]] said BTB and said recent entry queue by examining the contents of [[the]] said recent entry queue for such duplicate entries prior to a write into said BTB and said recent entry queue and in addition for allowing a decode to be delayed ~~stalled~~ by a defined ~~amount~~ number of cycles such that a branch of interest can be delayed from decoding in order to allow a given entry in [[the]] said BTB to be detected in time for future decodes of said branch of interest.

29-30 (Canceled)

31. (Previously presented) The program product of claim 28 comprising code for making the recent entry queue fully associative for reading.

32- 39 (Canceled)

40. (Originally Presented) The program product of claim 39 comprising code for fast indexing recently encountered branches.

41. (Previously presented) The program product of claim 39 comprising code for searching the complete recent entry queue to block duplicate BTB writes.

42-46 (Canceled)

46. (Currently Amended) The program product of claim [[28]] 58 comprising code for delaying decode until a fixed number of cycles.

47. (Originally Presented) The program product of claim 46 comprising code for delaying decode until the BTB predicts a branch.

48. (Originally Presented) The program product of claim 28 comprising code for staging writes to the BTB in the recent entry queue.

49. (Previously presented) The program product of claim 48 comprising code for delaying a write and placing the write in the recent event queue.

50. (Previously presented) The program product of claim 49 comprising code for detecting a predicted branch while its BTB write is temporarily staged in the recent entry queue.

Add the following claims which replace former claims 4, 19, 20, 30, 34, 38, 39 and 46 which were rejected as non compliant.

Claim 51 (new): The method of claim 1 wherein said recent entry queue is fully associative for reading.

Claim 52 (new): The method of claim 1 comprising delaying decode until a fixed number of cycles.

Claim 53 (new): The method of claim 1 comprising delaying decode until the BTB predicts a branch.

Claim 54 (new): The program product of claim 28 further comprising code for organizing the recent entry queue as a FIFO queue.

Claim 55(new): The program product of claim 28 further comprising code for writing an entry into the recent entry queue when the entry is written into the BTB.

Claim 56(new): The program product of claim 28 comprising code for creating hit detect logic to support the associativity of the BTB.

Claim 57 (new): The program product of claim 28 comprising code for using a subset of the recent entry queue as a subset of the BTB.

Claim 58 (new): The program product of claim 28 comprising code for delaying decode until a fixed number of cycles